

Spikesort (A MATLAB-based spike sorter for NEV files)

Maintained by the Smith Laboratory (www.smithlab.net). If you use Spikesort and find it useful (or if you have problems), please drop us an email (matt@smithlab.net).

Authors:

Ryan Kelly - wrote the original version of Spikesort in 2006 while he was a PhD student at CMU

Matthew Smith - worked with Ryan Kelly on the original version, and led development since 2011

Adam Snyder - extensive development of Spikesort from 2012-2018 while at Pitt/CMU as a postdoc

Shruti Vempati - contributed bug fixes and improvements in 2016-2017 while she was a BioE undergrad at Pitt

Xiaohan Zhong - cleaned up the code and added features in 2018 prior to the GitHub initial release

Publishing data using Spikesort: If you use it in work you're going to publish, we would appreciate it if you would acknowledge us. You could put the software name and URL in your paper, or you could cite one or both of these references:

- Kelly RC, Smith MA, Samonds JM, Kohn A, Bonds AB, Movshon JA & Lee TS (2007) Comparison of recordings from microelectrode arrays and single electrodes in the visual cortex. *Journal of Neuroscience*, 27: 261-264
- Kelly RC (2010) Statistical Modeling of Spiking Activity in Large Scale Neuronal Networks, Ph.D. Thesis, Carnegie Mellon University

DISCLAIMER: Spikesort is intended for sorting spike waveforms in NEV files (native format of Blackrock and Ripple hardware). This software modifies the NEV files and rewrites the sort codes for the channels. The authors of Spikesort take no responsibility for any errors in the software or in its use with your data. Remember that Spikesort modifies the NEV file, so you should have a backup before running this software. You should check that the sorting did what you thought it did with your NEV file. **YOU ARE RESPONSIBLE FOR YOUR OWN DATA!** That said, Spikesort continues to undergo development and has been used extensively. Bugs have existed in the software and will exist again, but it is relatively robust and mature.

Main File List

genSpikeMapColor.c + mex-compiled files - take waveforms and produce an image shown in the GUI

genSpikeMapColorMatlab.m - [NOT USED] this is a slower Matlab-native code version of *genSpikeMapColor.c*

getSNR.m - calculate SNR (signal-to-noise ratio) - see Kelly et al 2007 (*Journal of Neuroscience*, 27: 261-264)

loadSort.m - load the saved sort file

manageTempFiles.m - manage temporary Spikesort files

readSampleWaveforms.m - load a subset of spikes for sorting

readWaveforms2.m - read the waveforms, times and units

readWaveforms2_timer.m - uses a timer function so that it can run in the background

readWaveforms2_timer_sparse.m - does a sparse (limited) read of waveforms for speed

rubberbandbox3.m - draw a rubber band box and return the start and end points (for selecting spikes)

rubberBandMotionFunction.m - window motion callback function

saveSort.m - save a sort to a specified location

spikesort_gui.m - define the operation callbacks

spikesort_nevscan.m - read the NEV file & get file information (called when a set of files are loaded)

spikesort_write2.m - called to write the sort codes to the NEV file

spikesort.m - spikesort main function

Utility File List (in some cases these are not standalone, and call other Spikesort functions)

chanPlot.m - plot the spike waveforms for specified channel(s) with various plot options

justSNR.m - return channels list, sort codes, SNR values and spike counts (and average waveform)

moveToSortCode.m - write a list of sort codes into a NEV file

plx2nev.m - take a PLX file and convert it to NEV file with the same file name but a NEV extension

plxSortFromNEV.m - take a NEV-converted PLX file & write sort codes back into the PLX file

waves2nev.m - take waveforms, channels, sort codes, times and write them into a NEV file

Notes

Mex Files: Spikesort uses a Mex-compiled C function (genSpikeMapColor.c) to speed up generating the waveform image for display. This function is currently compiled for several platforms. If you need to recompile, go to the Spikesort directory and type "mex genSpikeMapColor.c". Alternatively, there is a matlab-native version (genSpikeMapColorMatlab.m), but it is much slower.

NEV files: Spikesort works on NEV files, the native data format for Blackrock and Ripple hardware. If you have data from another recording setup, you can export your waveforms and convert them to a NEV using the waves2nev.m function supplied in the Spikesort directory. Also plx2nev.m converts Plexon PLX files to NEV.

Performance: Spikesort creates temporary sort files of in-progress sorts. When you first run Spikesort, it will ask you where to store these files (you can change the location in the GUI). It will create a "spikesortunits" folder that contains ch*.mat and hist*.mat files. These are the cached waveforms and the sorting history for each channel. It cleans up that directory and creates a tempsort_*.mat file on exit. The speed of Spikesort will be affected strongly by the disk access speed of the NEV file and also of this spikesortunits directory.

Spikesort GUI

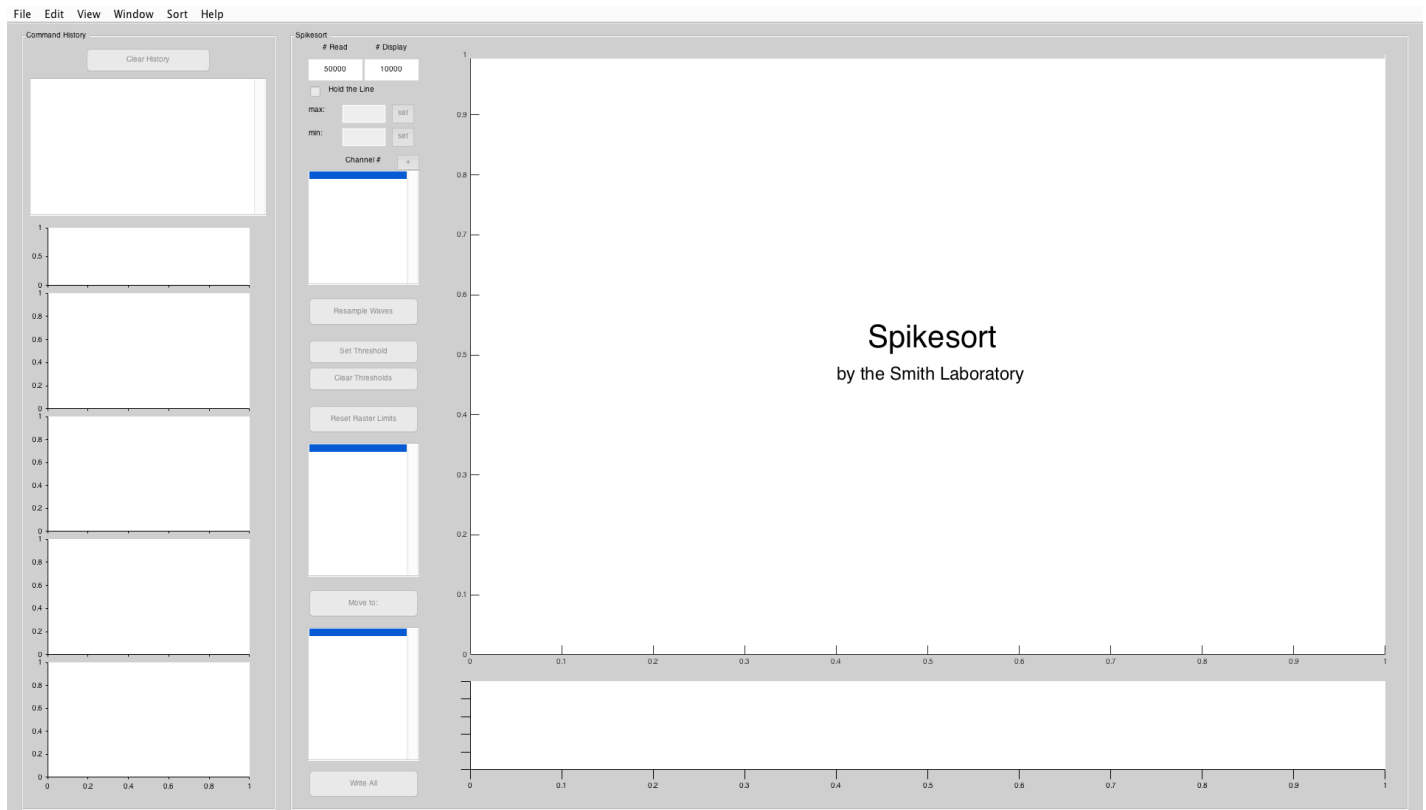


Figure 1

Opening Spikesort and loading data (Before getting started add the spikesort directory to your path)

Method 1: Using the GUI

- Open Matlab and type “spikesort”
- Loading Options: Under file (Figure 2) change loading options before loading data (Sparse load, Background load). Click on the text and a check mark will appear next to the selection. Read more about loading options below.
- Load Data: Under file select “Load data”. Load single or multiple files to sort (Figure 3).
- You can use shift or control modifiers to select multiple files
- Files will always be sorted alphabetically from the list of files selected.

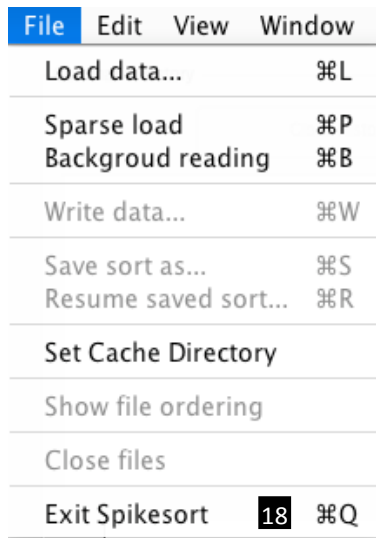


Figure 2

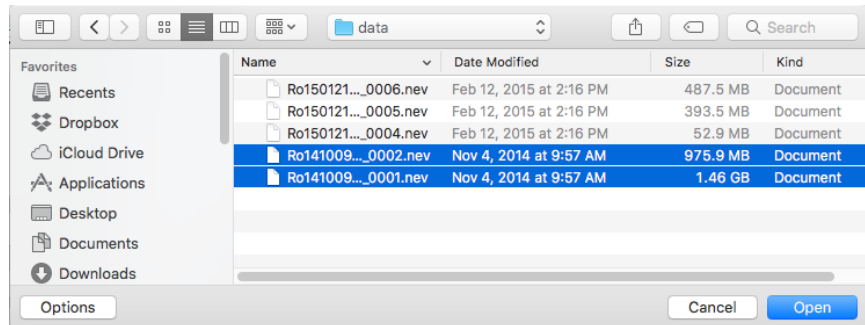


Figure 3

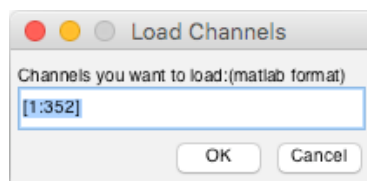


Figure 4

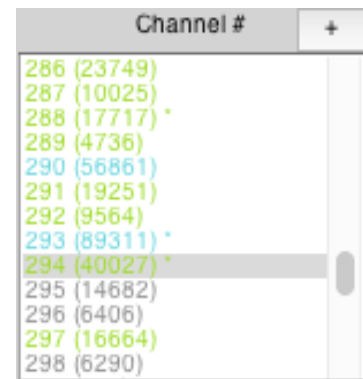


Figure 5

Method 2: Using the command line

- All the loading options in the GUI are available at the command line.
- Spikesort accepts one or multiple files at once. All file inputs must be cell arrays. It also accepts optional input arguments 'doSparse', 'doTimer' and 'noSort'.
- 'doSparse' enables the sparse read capability. It can be logical input or positive integer which specifies the subset number of waveforms when loading. Default is false.
- 'doTimer' enables the background reading capability. It can be logical input. Default is false.
- 'noSort' disables the sorting of the cell array of input filenames, so you can specify a particular order. It can be logical input. Default is false (which means the input files will be sorted alphabetically). (You can find the order of files by clicking 'Show file ordering' under 'File' Menu)
- Examples:

```
spikesort({'path/animal_date_task_0001.nev'});  
spikesort({'path/animal_date_task_0002.nev','path/animal_date_task_0001.nev'},'noSort', true);  
spikesort({'path/animal_date_task_0001.nev'},'doSparse',1000,'doTimer', true);
```

Loading Options:

- Load Data: After selecting files a pop-up window (Figure 4) will ask which channels you want to load. In the “channel #” window in Spikesort the loaded channels will be in green text and unloaded channels will be in gray text (Figure 5).

- After the initial loading, channels that haven't been loaded can be loaded by double clicking the channel.
- If you click multiple channels they will load in the order you clicked them. This will be written out in text above the spike window (best practice is to wait until a channel is done loading before trying to load another one).
- If you click the "+" button, you have the option to select a range of channels to load (Figure 4).
- Sparse Load: load only specified number of units, speeding up the loading time
 - When checked in the GUI all future loading (double-clicking, etc.) will be sparse and show as blue text (Figure 5).
 - Default is set to 50000, this needs to be changed before loading the initial set of data or it will be locked in for the remainder of sorting.
 - Use 'doSparse' as an argument at the command line as well. The default is 50000 but can be by specifying it in the input.
 - `spikesort({'datafiles.nev'}, 'doSparse', 50000)`
- Background Load: loads data in the background so that you can sort other channels while it is loading.
 - Use 'doTimer' to add this feature at the command line
 - `Spikesort({'datafiles.nev'}, 'doTimer', true)`
- Double clicking grayed channel: this will automatically load in the background. If sparse load is checked the files will be sparse loaded.

Setting Cache Directory

The first time Spikesort is opened it will ask you where you want to store your files. This is also where temporary files will be stored. If you ever want to change where that directory is just go to file, then click "Set Cache Directory" (Figure 2).

Loading temporary sorts

Anytime Spikesort is closed and data hasn't been written or saved a temporary sort file is saved to the cache. When opening a datafile that has a temp sort, one of two pop up windows will prompt you to either open a temp sort, open the NEV, or load NEV and delete temp sort (Figure 6). If more than one temporary sort file is found it will ask if you want to pick one of them (Figure 7). A second window will pop up if you have more than one sort to choose from (Figure 8).

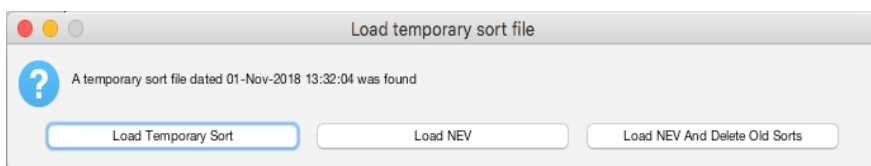


Figure 6

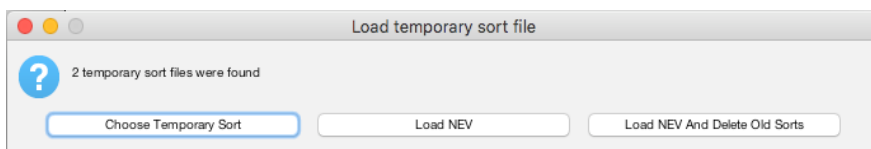


Figure 7

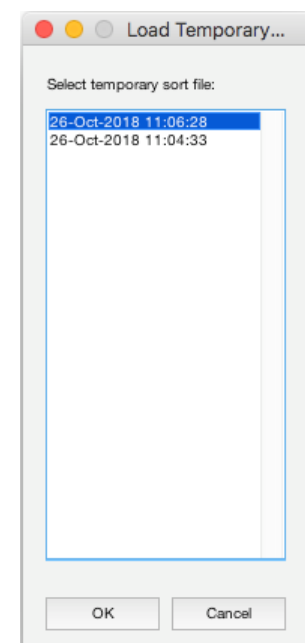


Figure 8

Sorting Options

Note: Automatic sort options are under the 'Sort' menu (currently under development).

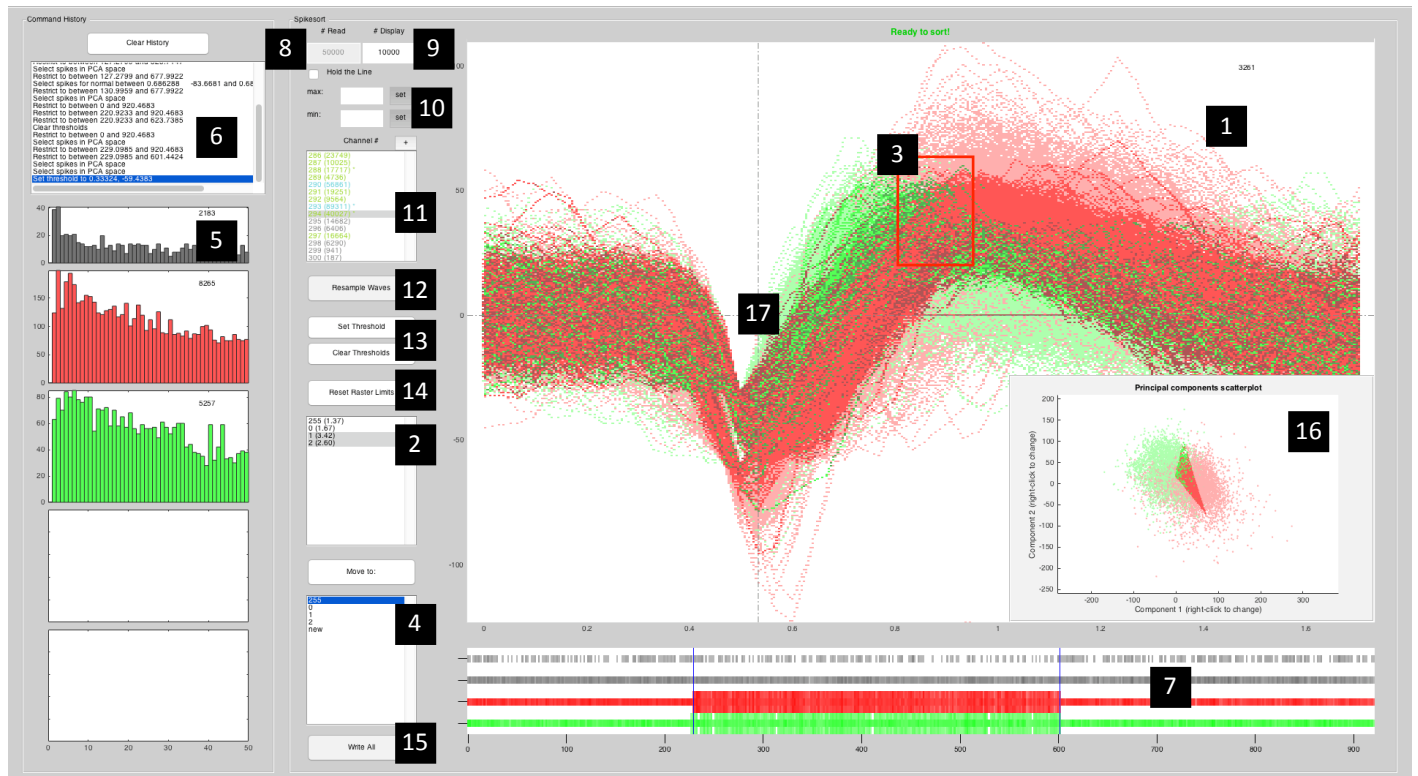


Figure 9

Spikesort GUI Introduction

- Main sorting window.** Waveforms from selected sortcodes (see [2]) are displayed here for manual sorting. The x-axis measures time (in milliseconds) and the y-axis measures voltage (in millivolts).
If no waveforms are selected, (given by the integer in the top-right corner of the window)
 - Left click and drag to create a rubber band box that selects all waveforms passing through it.
 - Right click and drag to create a rubber band box that selects all waveforms not passing through it.*If waveforms are selected,*
 - Left click anywhere to deselect all waveforms.
 - Left click and drag to select a new set of waveforms, as above.
 - Right click and drag to create a rubber band box that deselects the subset of selected waveforms that passes through it.*Relevant keyboard shortcuts (Only active when "current axes" is the main sorting window)*
 - Up/Down arrow keys: Change channel, see [11]
 - Left/Right arrow keys: Change sortcode, see [2]
- Selected sortcodes.** Waveforms from sortcodes selected in this listbox appear in the main sorting window. Sortcodes 1-254 are intended to store spiking waveforms, while 0 and 255 are to store noise. For each sortcode, the signal-to-noise ratio (SNR) of all of its waveforms is given in parenthesis.
Relevant keyboard shortcuts (Only active when "current axes" is the main sorting window)
 - Left/Right arrow keys: Change sortcode*Note: Actions in the main sorting window operate only on selected sortcodes. Multi-select supported.*
- Rubber band box.** Select/deselect waveforms in the sorting window. See [1].
- Selected waveform destination.** The "Move to:" button moves all spikes selected in the main sorting window into the sortcode highlighted in the listbox beneath it. If no waveforms are selected or waveforms are moved to the same sortcode, nothing happens. If waveforms on one or more sortcodes are selected, they are all moved to the destination sortcode and remain selected in case of user error. For more on handling user errors, see [6].
Relevant keyboard shortcuts (Only active when "current axes" is the main sorting window)
 - 0, 1, 2, 3, ...: Move selected waveforms into sortcode #.
 - (Windows) Delete: Move selected waveforms into sortcode 255. (Mac: Fn+Del)
- ISI histograms.** Distributions of inter-spike intervals (ISIs) for each sortcode. Useful secondary factor to consider when making sorting decisions. The x-axis measures time (in milliseconds) and the y-axis is a count of waveforms.

6. **Command history.** Full sequence of all sorting commands executed on the present working channel. Double clicking any previous command will undo all commands that follow it in the menu; pressing the "Clear History" button will undo all commands, resetting the channel.
Relevant keyboard shortcuts (Only active when "current axes" is the main sorting window)
 - (Windows) Backspace: Move backward one command in the history (Undo). (Mac: Del)
7. **Spiking raster.** Spiking raster plots of waveforms on each sort code across the length of the data. Multiple datasets being sorted simultaneously are put back-to-back, separated by a black vertical line (not shown). Only spiking waveforms that occurred within the blue vertical bars are shown in the main sorting window.
To restrict the waveform viewing window,
 - Left endpoint: Left click once to bring up a crosshair. Left click again to move the left endpoint (left blue vertical bar).
 - Right endpoint: Right click once to bring up a crosshair, again to move the right endpoint (right blue vertical bar).
 - While the crosshair is active, press (Windows) Backspace to reset the viewing window to the fulltime interval. (Mac: Fn+Del)

Note: Actions in the main sorting window operate only on waveforms within the waveform viewing window.
8. **Number of sparse read waveforms.** Default 50000. Number of waveforms read when sparse read selected. If the total number of waveforms for selected electrode channel is smaller than the number of sparse read waveforms, all waveforms will be read instead of sparse read. It can only be changed before loading a file.
9. **Number of display waveforms.** Default 10000. Number of waveforms shown in the main sorting window. Larger numbers risk slowing the GUI. It can be changed anytime.
Note: ALL WAVEFORMS, not simply the visible ones, on selected sort codes are affected with sorting commands.
10. **Constrain axes limits.** Place constraints on the y-axis of the main sorting window. Default behavior changes axis limits per channel to best display the waveforms within.
In the case of high-voltage noise waveforms on a channel, fill in some voltage in the "max:" box to set a maximum and reselect the channel from the channel selection box (see [11]).
If "Hold the Line" is checked, axes limits remain constant across sortcodes. If not, axes limit automatically change to best display the selected sortcodes.
11. **Channel selection window.** Selected electrode channel is active in the GUI workspace. Each loaded channel has its own independent waveforms, sortcodes, and command history.
Each list item gives channel number, followed number of waveforms on the channel in parentheses and an asterisk (*) if unwritten changes have been made. The list item is shown in green when the channel is fully loaded, in cyan when the channel is sparse read and in grey when the channel has not been loaded yet (Notification will be shown in main sorting window).
Relevant keyboard shortcuts (Only active when "current axes" is the main sorting window)
 - Up/Down arrow keys: Change present working channel
12. **Resample waves.** Resample the waveforms showing in the main sorting window from the total read waveforms.
13. **Set threshold/Clear threshold.** Shift waveforms horizontally according to a selected y-axis (voltage) threshold. "Set Threshold" opens a crosshair for the main sorting window. "Clear Thresholds" clears all existing thresholds, if there are any.
When setting threshold (with left click in main sorting window),
 - Vertical placement of crosshair sets voltage threshold for any waveform that crosses it.
 - Horizontal placement of crosshair determines alignment point for waveform shift.
 - The first point that a given waveform crosses the set threshold is translated horizontally to be placed on the alignment point.
14. **Reset raster limits.** Reset the spiking raster plots to default.
15. **Write all.** Write all sorting commands to .nev file.
*Note: Save as tempSORT *.mat file before making changes to original .nev file(s). Rest assured, the changes may be reversible, and the program will ask if you are completely certain that you wish to write the changes before doing so.*
16. **PCA scatter.** Click "Show PCA scatter" under "View" to get principle component analysis scatter plot. Each spike is represented by 5 components. Right click x-axis or y-axis to change the components. Each point on the scatterplot corresponds a spike on main sorting window. Left click on the plot to get "+" arrow, outline the points and right click to confirm, the selected points(spikes) have relatively dark color than the unselected ones.
17. **Show threshold.** Click "Show threshold" under "View" to draw horizontal lines of low threshold and high threshold, also draw a vertical line of timestamp of threshold location, which is where waveforms align when click "Set Threshold".
18. **Exit Spikesort.** Exit Spikesort software.

Note: Spikesort creates temporary sort files of in-progress sorts. In your current working directory in Matlab, it will create a "spikesortunits" folder that contains ch.mat and hist*.mat files. These are the cached waveforms and the sorting history for each channel. It cleans up that directory and creates a tempSORT_*.mat file on exit by clicking "Exit Spikesort" or "x".*